REPLACEMENT PARAGRAPHS FOR THE SPECIFICATION
SHOWING AMENDMENTS MADE THERETO

Please replace the paragraph at page 2, lines 10-22 with the following
replacement paragraph (indicating amendments thereto):

According to embodiments of the present invention, methods and processors for multiplying two maximally negative fractional numbers to produce a 32-bit result are provided. This type of multiplication may be executed using a 16-bit DSP multiplier and produce a 32-bit result. The identification of a multiplication operation employing two maximally negative 16-bit fractional numbers enables manipulation of processing to correct a maximally negative result and produce a maximally positive result. Negate logic with a control block examines results produced by the 16-bit DSP multiplier and determines whether there are a combination of bits signifying the multiplication of two maximally negative 16-bit fractional numbers. The determination of the required bit combination initiates negate processing for correcting the results. This type of multiplication operation utilizes [[of]] a 16-bit DSP multiplier to ~~produces~~ **produce** accurate 32-bit results when the multiplication of two maximally negative fractional ~~number~~ **numbers** ~~occurs~~ **occur** as well as reduces the overall cost of the processor.

Please replace the paragraph at page 5, lines 2-14 with the following replacement
paragraph (indicating amendments thereto):

According to embodiments of the present invention, methods and processors for multiplying two maximally negative fractional numbers to produce a 32-bit result are provided. This type of multiplication may be executed using a 16-bit DSP multiplier and produce a 32-bit result. The identification of a multiplication operation employing two maximally negative 16-bit fractional numbers enables manipulation of processing to correct a maximally negative result and

produce a maximally positive result. Negate logic with a control block examines results produced by the 16-bit DSP multiplier and determines whether there are a combination of bits signifying the multiplication of two to maximally negative 16-bit fractional numbers. The determination of the required bit combination initiates negate processing for correcting the results. This type of multiplication operation utilizes [[of]] a 16-bit DSP multiplier to ~~produces~~ **produce** accurate 32-bit results when the multiplication of two maximally negative fractional ~~number~~ **numbers** ~~occurs~~ **occur** as well as reduces the overall cost of the processor.

Please replace the paragraph at page 8, lines 13-22 with the following replacement paragraph (indicating amendments thereto):

The data memory and registers 120 are volatile memory and are used to store data used and generated by the execution units. The data memory 120 and program memory 105 are preferably separate memories for storing data and program instructions respectively. This format is [[a]] known generally as a Harvard architecture. It is noted, however, that according to the present invention, the architecture may be a Von-Neuman architecture or a modified Harvard architecture which permits the use of some program space for data space. A dotted line is shown, for example, connecting the program memory 105 to the bus 150. This path may include logic for aligning data reads from program space such as, for example, during table reads from program space to data memory 120.

Please replace the paragraph at page 9, lines 1-5 with the following replacement paragraph (indicating amendments thereto):

Referring again to Fig. 1, a plurality of peripherals 125 on the processor may be coupled to the bus [[125]] **150**. The peripherals may include, for example, analog to digital converters,

timers, bus interfaces [[and]] <u>having</u> protocols such as, for example, the controller area network

(CAN) protocol or the Universal Serial Bus (USB) protocol and other peripherals.   The

peripherals exchange data over the bus 150 with the other units.

<u>Please replace the paragraph at page 9, line 21 to page 10, line 3 with the</u>

<u>following replacement paragraph (indicating amendments thereto):</u>

The W registers 240 are general purpose address and/or data registers. The DSP engine

230 is coupled to both the X and Y memory buses and to the W registers 240. The DSP. engine

230 may simultaneously fetch data from each [[the]] X and Y memory, execute instructions

which operate on the simultaneously fetched data, [[and]] write the result to an accumulator (not

shown) and write a prior result to X or Y memory or to the W registers 240 within a single

processor cycle.

<u>Please replace the paragraph at page 11, line 19 to page 12, line 2 with the</u>

<u>following replacement paragraph (indicating amendments thereto):</u>

Fig. 3 depicts a functional block diagram of a processor for multiplying two maximally

negative fractional numbers to produce a 32-bit maximally positive fractional result according to

an embodiment of the present invention. Referring to Fig. 3, the processor includes data memory

[[305]] <u>300</u> for storing data, such as two maximally negative fractional operands, used and

generated by the processor. The processor also includes registers 305 for containing .data that

the processor generates and employs during processing mathematical operation instructions. The

registers 305 may include a set of 16-bit W registers defined for mathematical operation

instructions. The set of W registers may contain data including an operand and an effective

address of an operand in data memory 300. The effective address of an operand in data memory

[[305]] <u>300</u> can be specified. as an address or an address plus an offset. The processor also

includes DSP unit 310, negate logic 325, fractional alignment logic 330, sign extension logic 335

and ~~accumulator~~ <u>accumulators</u> 340.

<u>Please replace the paragraph at page 12, lines 10-19 with the following</u>

<u>replacement paragraph (indicating amendments thereto):</u>

DSP logic 320 is activated upon the execution of mathematical operation instructions. In

this regard, when a mathematical operation instruction is executed control signals cause the DSP

unit to fetch operands from the registers 305 and the data memory 300. The control signals also

cause the DSP logic 320 to operate on the fetched operands to produce result outputs in

accordance with the instruction. The result outputs depend upon the instruction executed and the

source operands. The result outputs may correspond to a maximally negative result output. The

production of a maximally negative result output is based on the multiplication of two maximally

negative numbers, such as -1 * -1. After generating the result outputs, the DSP unit 310 writes

the result outputs into the correct registers 305, data memory 300, and ~~accumulator~~

<u>accumulators</u> 340.

<u>Please replace the paragraph at page 13, lines 12-22 to page 10, line 3 with the</u>

<u>following replacement paragraph (indicating amendments thereto):</u>

~~Control~~ <u>The control</u> block <u>of the negate logic 325</u> examines the two most significant

bits of the bits representing the result output and determines whether the two most significant

bits in the set of bits representing the result output have a particular bit combination. The two

most significant bits in the set of bits examined are the thirtieth and thirty-first bits for the set of

bits representing the result output. Upon determining that the thirtieth and thirty-first bits are one

and zero respectively, **the** control block **of the negate logic 325** generates a control signal to modify a negate control signal generated by negate logic **325**. The modified negate signals causes negate logic to perform a two's compliment operation on the result output correcting .the maximally negative result output to a maximally positive result output. The error introduce by correction of the maximally negative result output is nominal, and more, specifically one least significant bit of the result output.